

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 2.0 | 08/02/20 | Peta Clarke | Revised the evaluation |

**Instructions**

Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

## [Executive Summary](#_sbfa50wo7nsh)

Currently the game *Draw It or Lose It* is being served on Android devices only. Since the goal is to develop a web-based version of the game that can run on different platforms, it is important to keep in mind that with this software design, there will be unavoidable pros and disadvantages for each OS system or mobile device while implementing the web based version. The major issue with this design is choosing which route to take, whether it is client-side or server-side development. For this project, I would recommend using the client-side route and here are my reasons why. Server-side development works great when designing static pages. Since Draw It or Lose It will have a lot of user site interactions, it is best to go with client-side. A critical piece of info that should be known is that the initial rendering load might be a bit slow, but every subsequent load will be faster.

## [Design Constraints](#_2et92p0)

The design constraints for developing this game in a web based distributed environment is having to work around each platform’s server/client-side disadvantages. It also depends on the time and financial budget set aside for the development.

## [System Architecture View](#_ilbxbyevv6b6)

N/A

## [Domain Model](#_8h2ehzxfam4o)

****

**Class Relation:** The Game, Team and Player classes all inherit from the Entity class. Entity is a Super class. All three classes have an “is a” relationship with Entity. Team has a “has a” relationship with players, since each team has players. Also (Games and teams) and (GameService and Games) has the same relationship GameService has Games, Games has Team and Team has Player. These are all examples of aggregation.

**Object-Oriented principles:** This design implements**,** Encapsulation, Polymorphism, Inheritance, Association and Aggregation. These object-oriented principles help to keep the application secure and the code reusable and more readable.

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Has flexible command terminal settings for the server  Offers a server-based deployment method where the site will be hosted.  The costs associated with the server licenses is $1000 for unlimited client licenses. They also offer it at 29.99 for a minimum of 20 licenses. | It has flexible terminal commands as well but is also less pricy  Offers a server-based deployment method where the site will be hosted.  The costs associated with the server licenses depends on which server features you want to implement. Servers can range from 1000 and up | Has more supported software than the other Operating Systems  Offers a server-based deployment method where the site will be hosted.  The costs associated with the server licenses depends on which server features you want to implement. Servers price can range from $500 to around $6000 | A major con is that the other Operating Systems’ specs surpass mobiles, but it performs its best when the server is not mobile.  Offers a server-based deployment method where the site will be hosted.  The costs associated with the server licenses depends on which server features you want to implement as well as incoming traffic. The prices are $60 to $3000 |
| **Client Side** | Medium expertise and time required.  Around the price of PC, ~ Medium  To ensure web browser platform compatibility a client server pattern is needed. There are multiple options of either buying a server(s) or leasing them. | Large expertise and time required. Low cost.  To ensure web browser platform compatibility a client server pattern is needed. There are multiple options of either buying a server(s) or leasing them. | Little expertise and time required. Around the price of Mac, ~ Medium  To ensure web browser platform compatibility a client server pattern is needed. There are multiple options of either buying a server(s) or leasing them. | Large expertise and time. Low cost  To ensure web browser platform compatibility a client server pattern is needed. There are multiple options of either buying a server(s) or leasing them. |
| **Development Tools** | FrontEnd: JavaScript, HTML and CSS  Backend: Ruby, Python, Java, PHP  These programming languages are all free, but I believe that there should be separate teams working on each client platform so that they can focus on any specifics, so that may cost additional money. | FrontEnd: JavaScript, HTML and CSS  Backend: Ruby, Python, Java, PHP  These programming languages are all free, but I believe that there should be separate teams working on each client platform so that they can focus on any specifics, so that may cost additional money. | FrontEnd: JavaScript, HTML and CSS  Backend: Ruby, Python, Java, PHP  These programming languages are all free, but I believe that there should be separate teams working on each client platform so that they can focus on any specifics, so that may cost additional money. | FrontEnd: JavaScript, HTML and CSS  Backend: Ruby, Python, Java, PHP  These programming languages are all free, but I believe that there should be separate teams working on each client platform so that they can focus on any specifics, so that may cost additional money. |

## Recommendations

1. **Operating Platform**:

Recommended Operating Platform: macOS, Windows, mobile

1. **Operating Systems Architectures**:

Windows: Consists of two components – kernel (full access to system resources and hardware) and user mode

macOS: Base layer is the Unix core (memory use and CPU data flow), Next is graphics system that contains QuickTime and OpenGL(contains interface graphics and the rendering of images), then the app layer which consists of four components(offers three environments for app to run). Finally, the user interface (provides visual interface)

mobile: end user, device applications.

1. **Storage Management**:

The storage device used for the mobile platform is an SSD (Solid State Drive) which runs faster than HDD. HDD (Hard Disk Drive) are typically used in macOS and Windows. The storage should be used to hold all the app’s code except for the images that it will render.

1. **Memory Management**:

Phones and computers can easily store large files without any issues but by implementing a cloud server, it would alleviate the memory management issues on the devices and it would make the app run faster and more effective on all platforms.

1. **Distributed Systems and Networks**:

This all will be accomplished using Human Machine Interface (HMI) through the Internet Protocol and mobile IP. Since all devices have an IP address, various platforms and devices can communicate with other devices through an internet network using a server/client setup.

1. **Security**:

Numerous security challenges exist relating to the distributing of an application over multiple operating systems. This can range from unauthorized users gaining access to sensitive data, to bugs that have not been patched or found on each device. One way to lessen unauthorized access is by implementing file permissions to prevent users from altering the file. Another way is to implement an Intrusion Prevention System that will alert security personnel when there is suspicious activity on the network.